[10191/4405]

**WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY**
**(Supplementary Page)**

International Application No. PCT/EP2005/050289

**1 Documents**

Reference is made to the following documents:

D1: US-A-5,872,977 (THOMPSON ET AL) February 16, 1999 (1999-02-16)

D2: U.S. 2002/040469 A1 (PRAMBERGER JOHANN) April 4, 2002 (2002-04-04)

**2 Objections pursuant to Article 6 PCT**

2.1 The term "configuration" or "configuration data file" used in the claims leaves open whether the purpose of the method is a configuration in the sense of the parameterization of an application in operation, or in the sense of a configuration on the level of the source code during the creation of the application ("build management").

However, the description seems to indicate that the method is used for configuring on the level of the source code during the creation of the application (see, e.g., page 10, lines 28-33; Figure 1: *file_2.h, file_2.c*).

Therefore, the following argumentation is based on the assumption that the term "configuration" has the aforesaid meaning.

2.2 The step

> automatic generation of at least one implementation-dependent configuration data file as a function of the configuration data stored in the configuration data container;

does not permit one skilled in the art to determine which technical features are necessary to convert the implementation-independent configuration data, stored beforehand in the configuration data container, into implementation-dependent configuration data. According to the description, additional data like, for instance, hardware properties of a target platform seem to be used for this purpose (see also Claim 5).

Therefore, the following argumentation is based on the assumption that the at least one implementation-dependent configuration data file is generated as a function of the configuration data stored in the configuration data container **and** at least one hardware property.

## 3 Objections pursuant to Article 33(2) PCT

3.1 Document D1 is considered to be the most proximate related art with respect to the subject matter of Claim 1. It discloses (the references in parentheses relate to this document):

> A method for configuring **[a]** computer program including at least one functional unit, characterized by the following steps:

creation of at least one implementation-independent configuration data file and/or alteration of information filed in the at least one implementation-independent configuration data file (column 3, line 1: "*...platform independent build file...*").

automatic set-up and/or automatic update of configuration data, stored in a configuration data container, as a function of the information filed in the at least one implementation-independent configuration data file (column 3, lines 7-10: "*...internal data structure having the information necessary for forming the platform-dependent makefile ...*");

automatic generation of at least one implementation-dependent configuration data file as a function of the configuration data stored in the configuration data container (column 3, lines 10-11: "*...uses the information in this data structure to form the platform dependent makefile ...*");

automatic configuration of the at least one functional unit as a function of information filed in the at least one implementation-dependent configuration data file (column 4, lines 41-43: "*... automated means ...*"; column 7, lines 61-65; column 4, lines 29-30: "*... platform specific information...*").

3.2 The subject matter of Claim 1 is therefore not novel.

3.3 The subject matter of Claims 11, 13, 14, 15 and 16 corresponds, in the sense of device features or product features, to the subject matter of Claim 1. The objections

with regard to Claim 1 therefore hold true in the same way for Claims 11, 13, 14, 15 and 16.

3.4 The feature of Claim 5, to generate at least one implementation-dependent configuration data file as a function of at least one property of hardware on which an installation of at least a part of the configured computer program is to be made possible, is disclosed in document D1 (column 4, lines 29-30: "... *platform-specific information ...*").

## 4 Objections pursuant to Article 33(3) PCT

4.1 The feature of Claim 2, to automatically generate at least one item of dependency information that describes a dependency on at least two configuration data present in the configuration data container, and to generate the at least one implementation-dependent configuration data file as a function of the at least one item of dependency information, was already used in a similar connection for the same purpose; see, e.g., document D2 (page 3, paragraph [0037]: "... *The XSCML processor identifies ... logical dependencies ...*"). Therefore, without an inventive step, one skilled in the art would add this feature to the system of D1, in order to achieve the same effect.

4.2 The feature of Claim 3, according to which the computer program has a plurality of functional units, and a plurality of implementation-independent configuration data files is created, and each of the implementation-independent configuration data files is assigned to at least one functional unit, was already used for the same purpose in a similar technical connection, see, e.g.,

document D2 (page 9, paragraphs [0111] and [0112]: *EXAMPLE 3*). Therefore, without an inventive step, one skilled in the art would add this feature to the system of D1, in order to achieve the same effect.

4.3 The same holds true for the feature of Claim 4, according to which the computer program has a plurality of functional units, and a plurality of implementation-dependent configuration data files is generated, and each of the implementation-dependent configuration data files is assigned to at least one functional unit.

4.4 The feature of Claim 6, to generate the at least one implementation-dependent configuration data file as a function of the result of a plausibility check, represents an implementation detail which one skilled in the art would add to the system of D1 according to the circumstances, without an inventive step.

4.5 The same applies for the feature of Claim 7, to use the at least one hardware property for performing the plausibility check.

4.6 The feature of Claim 8, to automatically create a documentation, and the documentation describes the information filed within the at least one implementation-independent configuration data file and/or the at least one implementation-dependent configuration data file, was already used in a similar technical connection for the same purpose, see, e.g., document D2 (page 2, paragraph [0016]). Therefore, without an inventive step, one skilled in the art would add this feature to the system of D1, in order to achieve the same effect.

4.7 The feature of Claim 9, to create the at least one implementation-independent configuration data file in an XML-based format, was already used in a similar technical connection for the same purpose, see, e.g., document D2. Therefore, without an inventive step, one skilled in the art would add this feature to the system of D1, in order to achieve the same effect.

4.8 The feature of Claim 10, to automatically determine as a function of the configuration data, whether a functional unit included by the computer program is needed by the computer program, and to configure this functional unit only if the functional unit is needed by the computer program, is a feature common in build management systems.

4.9 Based on the remarks in Sections 3.4 and 4.1-4.9, the subject matter of Claim 12 does not satisfy the requirements of Article 33 PCT.